DigiClips Media Search Engine

sddec21-06

Tyler Johnson, Samuel Massey, Max Van de Wille, Max Wilson

Advisor: Ashfaq Khokhar

Our Client

- DigiClips, Inc. is a Colorado based media content analysis company
- Contacts:
  - Chairman: Bob Shapiro
  - Senior Software Engineer: Henry Bremers
- Constantly recording television news and radio in 210 markets
- Aiming to provide a search engine so that clients can search for keywords in broadcast recordings (mostly name, company names, etc.)

Presentation Schedule

- Project Overview
- System Design
- Demo
- Conclusion
- Q&A

Project Overview

- Problem Statement
- Functional Requirements
- Non-Functional Requirements
- Constraints and Considerations
- Project Milestones & Schedule

Project Overview

Problem Statement

- So far, only closed caption data is extracted alongside television recordings
    - No closed captions for radio broadcasts
- Closed captions data often misses words or phrases
- Any visible text shown on screen is also not transcribed
- Untranscribed data leaves gaps in the searchable content of a broadcast

Functional Requirements

- Speech-to-text must convert mono and stereo audio recordings into plain text
- Video-to-text must detect multiple fonts/styles of text on recording frames
- All system results must have appropriate searchable schemas
- All system results must have proper grammar and spelling
- All system errors must be alerted to the requesting service

Non-Functional Requirements

- System will be built without using any costly API/cloud resources
- System will be built with documentation to explain usage
- System should scale with increased quantity of data
- System should reliably output accurate data within the length of the original recording

Constraints & Considerations

Constraints:

- Cannot utilize certain paid APIs for speech-to-text or optical character recognition
- Developed program must be able to run on a relatively underpowered computer
- Must run quickly to query data within 24 hours of recording

Considerations:

- The output text must be indexed by timestamp so that it can be linked to a video segment
- Assuming video input will be high enough quality for accurate processing

Project Milestones & Schedule

Milestones:

- Complete speech-to-text system
- Complete video-to-text system
- Integrate the two systems into Driver

Schedule:

- April 2021 - December 2021
- April 2021 - December 2021
- September 2021 - December 2021

Evaluation Criteria

- Achieve 80% accuracy on speech recognition
- Achieve 70% accuracy on video text recognition
- Process speech-to-text for a video file within 75% of the file's length.
- Process video-to-text for a video file within the length of the file.

Challenges

- Speech-to-text on news and radio coverage
  - Multiple voices
  - Background music and noise
- Video-to-text on news broadcast
  - Inconsistent background makes locating text difficult
  - Font color can greatly affect accuracy

System Design

- Functional Decomposition
- Detailed Design
- Hardware & Software Platforms
- Test Plan
- Implementation

System Design

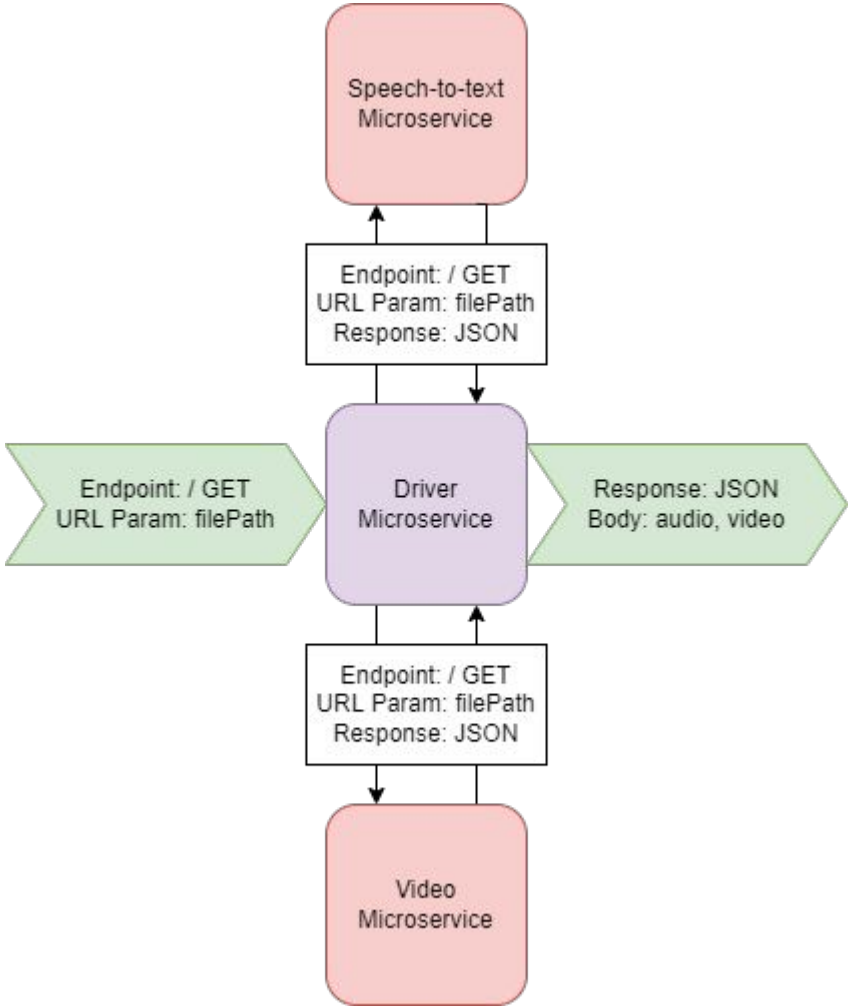
Functional Decomposition

- Detect speech in audio file and output transcript
  - Split input into chunks for individual processing
  - Process output for grammar and punctuation
  - Index output with timestamps
- Detect words and phrases shown on screen and output
  - Splitting video file into individual frames
  - Image pre-processing
  - Index text output with timestamps
  - Output filtering (spell-checking, duplicate filtering)

System Design

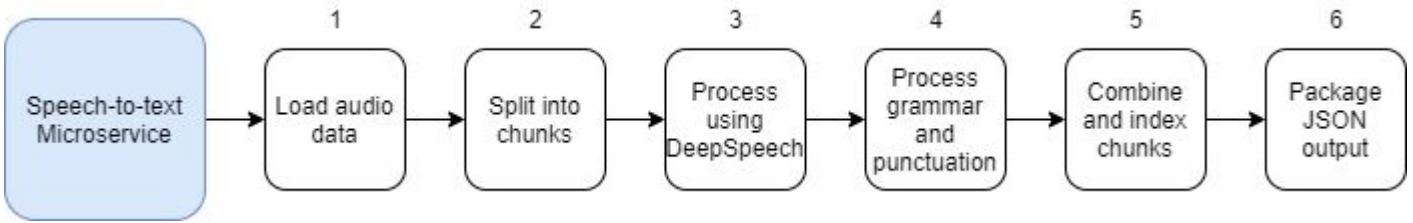
Detailed Design - Overall

- Microservices
  - Driver Microservice
  - Speech-to-text Microservice
  - Video-to-text Microservice



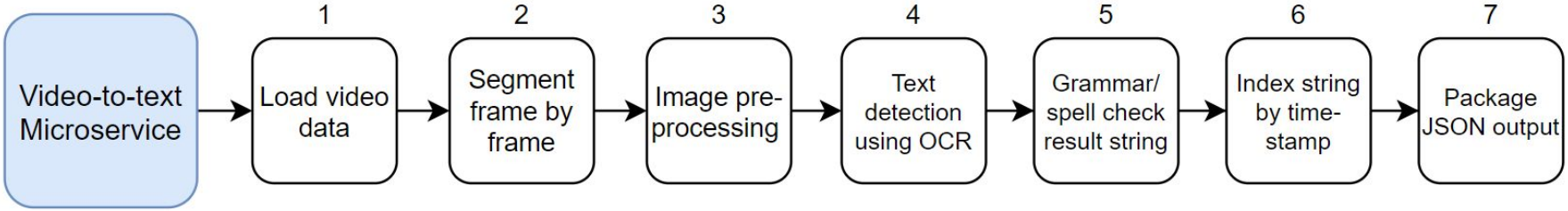
Detailed Design - Speech-to-text

- Load file into application
- Split into chunks on silence
- Run chunks through DeepSpeech model
- Add grammar and punctuation
- Combine chunks
- Output



System Design

Detailed Design - Video-to-text



- Perform pre-processing and text detection for every n frames
  - n is dependent on frames per second (fps) of input video
- Pre-processing includes binary mask, dilation, contour filtering
- Timestamp of frame (in seconds) can be found by calculating (frame number / fps)

System Design

Hardware & Software Platforms

Programming Language:

- Python

Frameworks and Libraries:

- FastAPI
- Tesseract OCR
- OpenCV
- Numpy
- DeepSpeech
- pydub
- punctuator2
- pyspellchecker

System Design

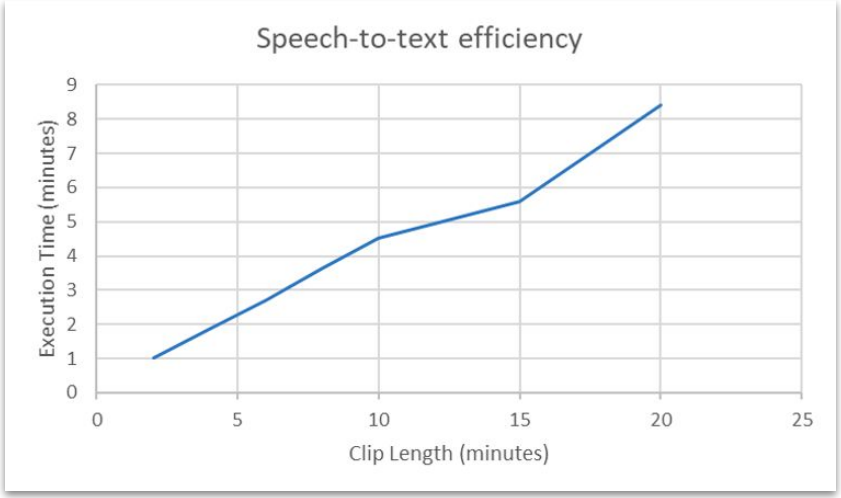
Test Plan

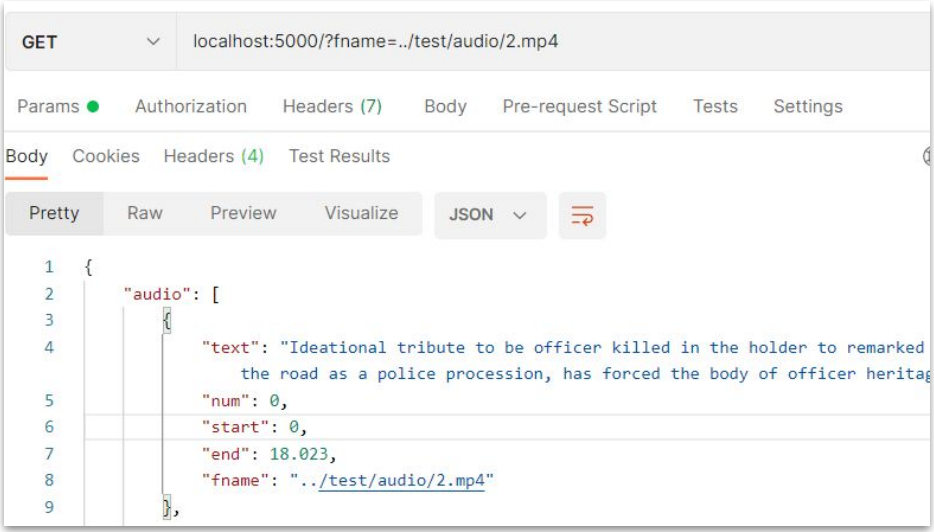
- Extremely time consuming to create test oracles for accuracy testing
- Many different ways to compare strings/sequences makes it difficult to get a true benchmark for accuracy
- Results
  - Peak accuracy of 82.5% accuracy for speech-to-text
  - Different comparisons provide different (but similarly high) percentages

```
C:\Users\Tyler\Box\College\CprE 492\text_testing>python testbench.py -t Transcribed.txt -g Generated.txt
Transcribed wordcount is: 268
Generated wordcount is: 233
Generated text is: 0.869 times the size of the transcribed text
Raw text ratio is: 0.825
Caseless text ratio is: 0.685
Grammerless text ratio is: 0.85
```

Implementation - Speech-to-text

- Implemented FastAPI microservice
- API accepts a file path
- Outputs JSON speech-to-text result





Example of using Postman to make requests to FastAPI service

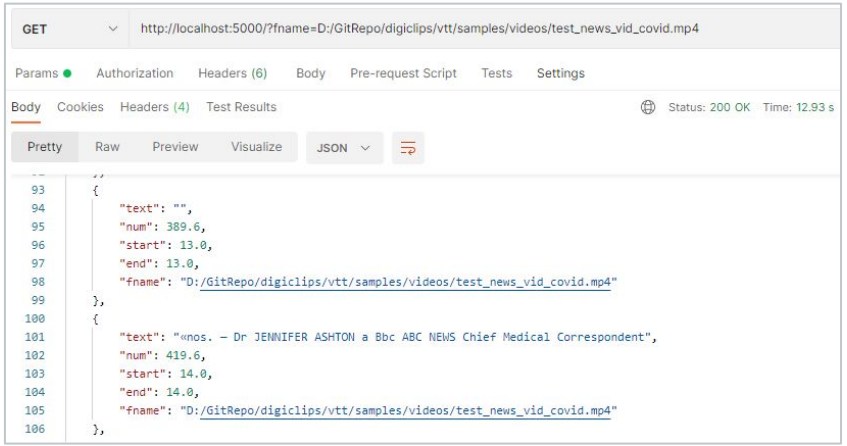
Implementation - Speech-to-text

- Break audio into ~20-second chunks
- Process each chunk in parallel using DeepSpeech
- Return formatted JSON results to Driver
- Test news recording resulted in 72% accuracy
    - Comparing manually transcribed clip to speech-to-text result

Implementation - Video-to-text

- Full API with image processing, output formatting
- Program accepts a file path to video
- Returns JSON-formatted list of identified strings with time-stamped start and end

Example output showing locations of identified text

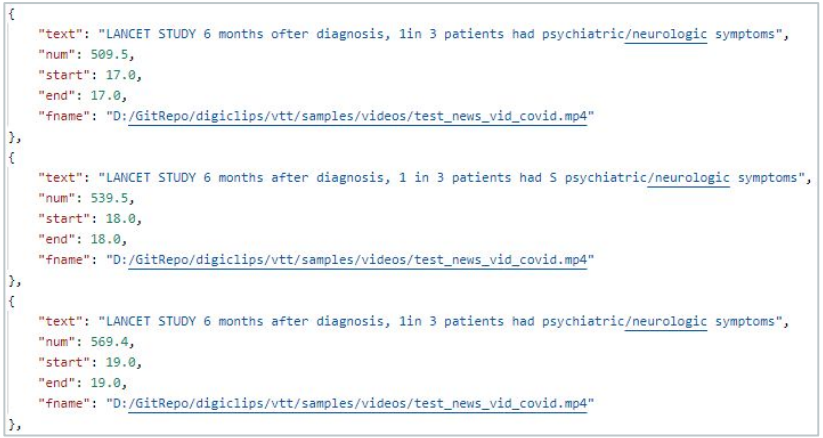


JSON output for previous figure with formatted timestamp

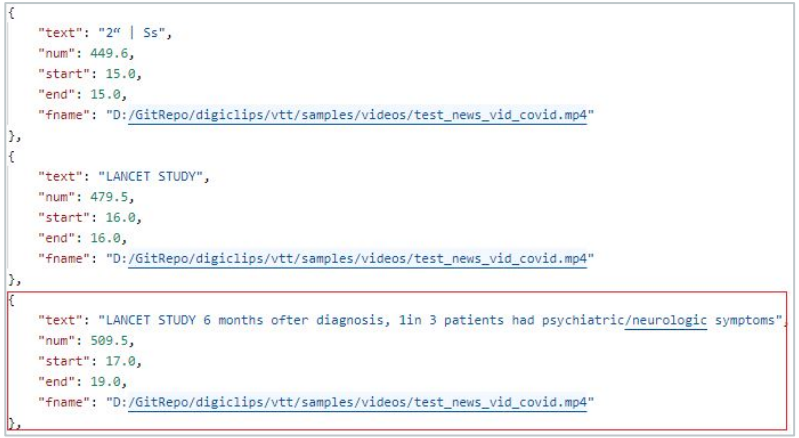
Implementation - Video-to-text

Implementation - Video-to-text

- Duplicate filtering combines similar neighboring strings into a start-end timestamp range
- Limits excessive repetition across multiple frame instances



Example output without duplicate filtering

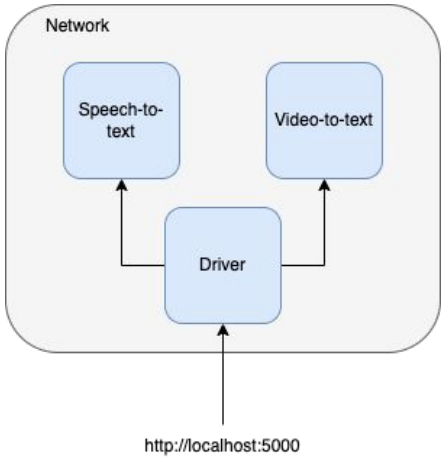


Example output with duplicate filtering

System Design

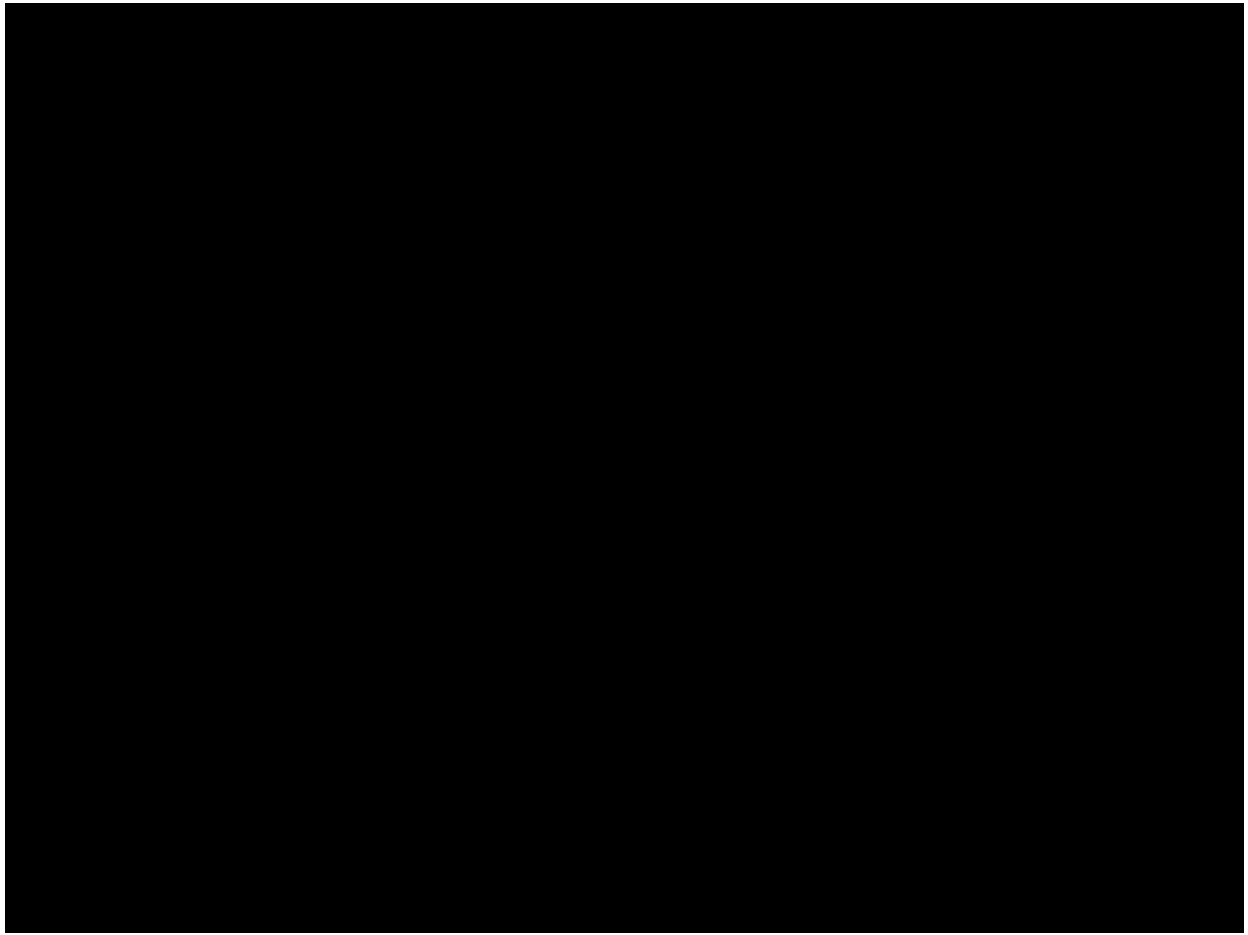
Implementation - Docker

- Portability
- Networking
- Ease of use



System Design

Demo



Conclusion

- Final Project Status
  - Functioning application
  - Documentation nearly completed for all subsystems
    - General operations manual completed for client use
  - Difficult to obtain true accuracy results but we feel confident the program is around the initial expected accuracy
- Future Improvements
  - More advanced parallel process management
  - Improvements to video and audio preprocessing to increase accuracy
  - Utilize GPU acceleration to speed up processing

Questions?

Appendix

- Market Survey
- Potential Risks & Mitigation
- Resource & Cost Estimates
- Gantt Schedule
- Task Responsibilities & Contributions

Appendix

Market Survey

- Very few existing implementations of speech-to-text and video-to-text on television
- Most similar applications differ in key areas
  - Performing processing on a live feed
  - Grammar and spelling is not a concern for output
  - Output is not formatted to be searchable
  - Usages are not time-sensitive

Appendix

Potential Risks & Mitigation

Risk	Probability	Mitigation
Speech-to-text processing inaccuracies	0.2	Extensively research speech recognition technology
Video-to-text service processing too intensive	0.5	Researching video OCR strategies and code optimization
Word misidentification	0.5	Testing throughout development Spell/Grammar output checking
System Integration	0.5	Containerized microservices limit integration issues

Appendix

Resource & Cost Estimates

Resources:

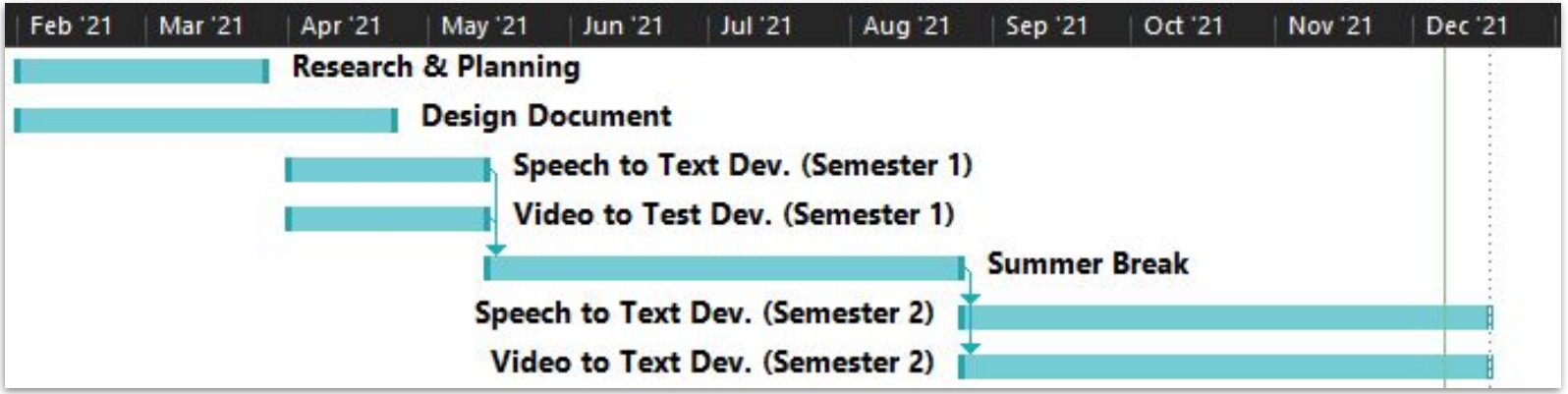
- No additional resources required to complete project

Cost:

- This project will not incur any costs

Appendix

Gantt Schedule



Appendix

Task Responsibilities & Contributions

- Tyler Johnson
  - Responsible for planning and implementing testing on project
- Samuel Massey
  - Responsible for assignment planning and research/work on speech-to-text
- Max Van de Wille
  - Responsible for documenting architecture changes and working on video-to-text
- Maxwell Wilson
  - Responsible as primary point of contact with client and working on speech-to-text